

Weekly Report

Period: 2018/10/22-2018/10/28

Reporter: Li Zongzhuang

绪论

1.1 可视分析

在人类的各类感觉器官中，人眼具有最强的信号处理能力。因此视觉是获取信息的最重要通道，超过 50%的人脑功能用于视觉感知，包括解码可视信息和思考可视符号等。但是，人类的视觉工作内存很小，在认知过程中常常需要依靠外部帮助。图片、图形等在过去两百年间是人类认知过程中最重要的工具。

可视化是人类在认知过程中产生的对事物的认知视图，易于理解并有利于知识交流。而信息可视化是让计算机绘制动态、交互的外部辅助视图，目的是增强人的认知能力。

随着数据规模的增加，直接呈现数据会导致显示空间不足、图元遮挡、绘制效率低等问题。另外，现代分析任务更加复杂，仅仅依赖可视化难以令人发现隐藏在数据背后的模式，迭代式交互分析是领域专家认识和理解数据的重要方法。

为此，可视化专家提出了可视分析概念，它是一种通过交互式可视化界面来辅助用户对大规模复杂数据集进行分析推理的科学和技术。可视分析不仅仅是可视化，它集成了可视化、人的要素（如交互、认知、感知、合作、表示和传播）和数据分析等。相对于可视化，可视分析整合了信息分析、地理空间分析和科学分析等技术，而且人的要素在人与机器的通信过程和决策过程中起到非常重要的作用。可视分析是人的智能（知识和经验）和机器智能（计算）多次协作的迭代过程。

可视分析有效融合了计算机强大的计算能力和人的认知能力，能够发挥二者的优势完成各自擅长的任务。因此，可视分析是交叉融合了可视化、人机交互、认知科学、数据挖掘、信息论、决策理论等研究领域的新研究方向。

可视分析需要合适的载体，而电力系统数据便是合适的载体。电力系统稳定是电力系统中的一个基本问题。电力工业因系统不稳定性会引起严重的停电事故和巨大的经济损失。

电网是电力系统的一种经典形式，被描述为通过发电机向用户传输电力的网络，一个结构良好的分配制度。一个电网是复杂的、动态的、多方面的系统。直到现在还没有一种可行的方法来纪录系统的情况。控制实验对实际的电力系统可能会造成意外的损失。这使得模拟计算性能分析成为唯一可以接受的方法。目前在实际应用中已经验证，模拟计算可以达到比较精确的测试。定期运行数据模拟，对实际事件进行推理，预防性问题分析。

在可视分析中所提供的数据驱动的分析工具，能够有助于分析人员从更多地角度、更深入地探索海量仿真数据中所蕴含的规律，使得电网运行分析具有更强地研究性，适应我国电网高速发展的现状。

1.2 电力系统仿真分析

仿真数据，尤其是暂态稳定计算在指导电网稳定运行中起着重要的作用。仿真数据通常是描述每个节点中每个变量的演变的一组时间序列。在模拟它的运行、故障和恢复过程中，不同形式的不稳定性伴随着它的演化过程而出现。数值模拟方法如李亚普诺夫函数及其变分已被证明是研究动态和瞬态不稳定性的独特手段。因而，分析人员要识别故障事件的发生，研究故障事件的类型和模式，并通过分析变量的时间序列来推断故障事件的原因，是非常麻烦的。

目前缺乏定量描述电力系统稳定状态的理论模型。传统的统计方法，如统计质量控制，提供了基于点的稳定性判断，但对于刻画全局趋势和模糊变化是低效的。

电力系统仿真分析广泛应用于电力系统运行、规划、设计等领域，是电力系统的一项基础性支撑技术。经过数十年的发展，我国基于机理模型的电力系统仿真计算技术已处于世界前列，已有 PSD、PSASP、ADPSS、PS-Model，以及全过程仿真软件等诸多广泛应用的自主知识产权优秀产品，可以基本满足大电网仿真分析的需求。

电力系统仿真分析中的“仿真”是指对电网元件进行数学建模，使用微分-代数方程组表达各个元件的外特性，通过求解这个方程组得到系统的状态变量值，从而掌握整个系统的情况。“分析”则是基于得到的仿真结果进行系统特性研究、控制措施制定等工作。在实际工作中，电网仿真分析可以看作是“提出假设”与“计算验证”的循环过程。

传统的仿真数据分析方式有着各种限制。首先是需要耗费大量人力，电网的日益复杂，使得人力的需求大大增加；其次是十分依赖专家的经验，分析的结论与专家的经验水平息息相关，能否发现问题和能否找出解决方法都取决于专家；再就是容易出现漏洞，由于主要依赖人工分析，难免发生错漏的现象；最后是无法控制误差。

先进的仿真计算技术必须要有先进的数据分析技术和人机交互方式才能充分发挥其作用。虽然目前的各类电网仿真软件均由结果可视化和报表功能，但其着重点仅限于数据的展示，对于数据规律的分析主要还是依靠人工进行只能观察很少的重要元件和断面。这极大地限制了对仿真结果和仿真能力的应用，使得许多在计算结果中蕴含的电网运行规律难以被发现。

基于这些问题，考虑用可视分析来帮助解决这些问题

电力系统的模拟数据包含多个变量及其演化。稳定性状态是所有变量演进的综合结果，要求对演进的细节和表达性视觉证据的可靠表示。

本文基于数值模拟数据，特别是暂态稳定数据，对电网稳定性进行了可视化分析。特别地，将统计质量控制（SQC）方法集成到具有创新视觉设计的稳定性时间的检测和研究中。对基于 SQC 的方法的增强促进了细粒度的稳定性判断，而细粒度的稳定性判断又通过富有表现力的视觉证据得到增强。

将 SQC 方法、可视化编码、领域知识和领域专家的经验无缝结合视为推进智能电网的新机会。我们的工作密切合作领域专家和经验丰富的电力仿真运营商。它们提供领域经验、程序、数据集和分析任务。最终设计和实现了一个综合的视觉系统，以支持多时间序列

的视觉探索和分析，识别和区分显著的不稳定模式。

支持电网的异常、不稳定和故障的勘探、识别和推理的可视化分析系统。视觉编码和 SQC 方法的系统集成，为研究可能影响系统稳定性的多个相关因素的联合效应提供可靠的视觉证据。一套用于数字时间序列的视觉设计，使分析人员具有对全球趋势、局部变化和多变量随时间变化的相关性的视觉检查的能力。

1.3 电力系统潮流计算

潮流，是电力系统的专业术语，指的是功率。电力系统在运行时，在电源电势激励作用下，电流或功率从电源通过系统各元件流入负荷，分布于电力网各处，称为电力潮流。电力系统是把很多的发电站、变电站、配电站、用户等由输电和配电线路连接起来形成的系统。电力(电能)是在发电站产生的，其中一部分在经过输配电线、变电站和配电站时损失掉，剩下的绝大部分最终被负荷所消耗。这样，从电的产生到被负荷消耗，流过哪一路输配电线，各节点电压是多少，这种计算就叫电力潮流计算或简称潮流计算。

电力系统潮流计算气计算。它的任务是根据给定的运行条件和网路结构确定整个系统的运行状态，如各母线上的电压（幅值及相角）、网络中的功率分布以及功率损耗等。电力系统潮流计算的结果是电力系统稳定计算和故障分析的基础。

潮流分析的目的在于获得电力系统的某个静态运行点，即所谓的潮流断面。由于电网是实时运行的，因此任何对于电网的计算和分析都需要从电网的某个断面出发来进行，例如系统发生故障前一刻的电网运行状态。也就是说潮流计算的结果除了用于潮流分析外，还需要用作其它计算分析输入数据的一部分。

潮流分析可以大体确定电网的一般运行状况（不考虑故障之类的冲击，这种冲击由暂态稳定分析考虑），比如各母线的电压。通过对数据的调整（按照一定要求），可以使潮流计算结果趋于合理，在满足相关要求的同时，也能为其它计算提供良好的基础。例如，如果发现电网内某些母线的电压偏低，可以通过投入无功补偿的方式使之升高。

潮流分析的结果可以用于作为电力系统实际运行的参考，即在实际电网调度中，正常情况下系统的控制边界都在已经计算过的潮流方式之内，比如系统的实际大运行方式不会超出之间计算的潮流大方式。当然，在具体下达调度命令时，调度员还是有一定的自由决定权的，只要系统运行在已经分析过的潮流方式框架之内就可以。

通常情况下，在一次电网分析工作中，需要进行多个运行方式的潮流分析并得到相应的潮流计算结果，诸如系统运行的大方式（重负荷）、小方式（轻负荷）等。基于这些结果，一方面可以确定电网在多个不同运行方式下的潮流分布情况，并采取相应措施；另一方面也能使暂态稳定、小扰动等计算可以基于不同的电网运行方式进行，获得更好地分析效果。

暂态稳定计算在电力系统规划和运行分析中占有重要位置。它不仅为规划系统的电源布局、网络接线、无功补偿和保护配置的合理性提供电力系统暂态稳定性的校核，为制定电力系统运行规程提供可靠的依据，而且可用于研究各种提高暂态稳定的措施并为继电保护和自动装置参数整定提供依据。

暂态稳定计算通常是由一个潮流计算结果开始，以故障作为输入，在某一时刻（每一步迭代）输入到计算中进行迭代，迭代的每一步代表一个时刻。在计算出结果后，研究院通常依靠观察攻角的变化来判断电网系统是否稳定。相比潮流计算，暂态稳定计算输出的物理量的个数更多，因而更有利于我们进行切入。

1.4 同行研发情况

到目前为止，监控电网的主要方式是通过对暂态稳定计算的结果进行计算和判断，从而掌握电网的运行状态。然而现有的方法仍有许多问题：

一是特征值都是人工提取的，关联关系太多，特征之间的关系不清楚；
二是现有的电网监控系统更加注重于特征值的计算与计算结果的展示，未能利用数据可视化的方法将数据统筹到一张图表中，使得监控起来繁杂难用；

三是现有的电网监控研究大多都在研究微网，即对城市内配电网的监控与判断，对于大尺度下的高压输电网，即从发电机到城市配电站之间的电网研究少之又少。

1.4 项目研究目标和内容

本项目依照现有的，通过模拟计算所得到的模拟电网特征数据，包括电网各关键节点（我们称之为“母线”）的电压值、频率值，以及模拟电网中的模拟发电机之间的相对功角值，对电网的暂态进行可视化编码与展示。

并且，我们会提供不同的图表格式，包括堆叠图，带状图等，对数据进行展示，并对数据进行统计与特征值提取，计算出一些合理的特征值，展现在项目中，供研究人员进行查阅。

同时，我们可以根据一些控制图计算算法计算出超出阈值的片段，以方便研究员进行后续分析，并可以将研究人员的分析结果记录到后端数据库中，以方便后人查看。

不仅如此，我们还会对大规模数据进行部分适配，以保证即使数据的规模比较大，整个系统的效率也能够合乎要求。

最后，我们会使用真实数据来验证我们设计的所有算法的合理性与准确性，并能给研究人员以提示。

数据抽象与任务分析

2.1 数据抽象

仿真数据记录了电力网络的运行状态。母线是一种由高导电率的铜（铜排）、铝质材料制成的，供“电站”或“变电站”使用以输送电能的产品。它具有汇集和分配电力的能力。网络中的每一个节点标识了一段母线，而一小部分节点标识了搭载在母线上的发电机。节点之间的连接是连接母线和供电区域的电力线。电气距离是指由母线间电力线路的正序阻和电抗所定义的物理量，指的是电网中成对节点的电气接近度。完善的数学模型可以从电力系统中抽象出节点链路结构，并构建一组包括数以千万计个变量的微分方程。通过改变迭代求解微分方程组时的初始计算配置可以获得一个求值集合，关于这个集合的可视化与分析已经引起了很大的关注。

在本文中，我们研究了用潮流计算模型对暂态稳定数据进行仿真计算。该仿真程序建立了基于电网基础设备（如母线，变压器和负载）的数学建模，将最常见的五类电气故障设置

为模型输入，包括母线接地、交流线路断开、发电机退出、线路接地和参数设置错误。仿真结果是一组能够说明电网设备随时间的动态变化状态的时变多变量数据。通常的，需要计算出多个时变物理量来描述电网设备状态的变化过程。本文作者选择了三个最有意义的特征来支持电网的稳定性分析：电压，频率和相对功角。频率和电压可以用来描述电网中所有节点的运行状态，相对功角则被用来描绘发电机对之间的状态。如果电网运行稳定，没有发生电气故障，功角值将保持不变；这个值的变化和波动只有在触发故障时才会发生。例如，当电气故障发生后，若电压值保持波动且不能收敛到一个恒定值，则电网很可能不稳定。类似地，这些特征值的任何异常变化都可以被认为是系统不稳定的暗示。由于电网的互联互通，特征的波动可能随时间而相互影响，使得电网稳定性分析更具挑战性。

总之，仿真数据记录了电气故障前后一系列物理特征的计算结果。采用传统的统计分析技术和折线图等可视化图表，电力分析人员可以观察到故障的发生，但很难推断故障事件的具体类型和模式，而如何描述不同变量间的时变关系也就变成了一项更具挑战的任务了。因此，本文合作者希望可以以一种直观客观的方式来刻画故障事件中不同变量随时间变化的相互影响。他们认为，这些特征的数值和统计值对于稳定性分析是有价值的，因为它们与整个系统的能量密度和分布有关，发明直观而全面的稳定性可视化分析方法具有重大意义。

2.2 任务分析

本文合作者从一开始就参与了这项研究，他们都对于如何对任务进行分析有丰富的经验。笔者也会与合作者们进行定期的讨论，并从讨论中相关的要求和各个方面的改进。每次讨论都会有对于电力系统分析有着丰富经验的领域专家在场。分析任务和要求在每次讨论中不断地更新改进，用来帮助系统进一步的设计。总体来说，本系统承担了以下几个任务：

判断电网是否稳定。电网的稳定性分析是非常重要的一层，也是电力系统分析上的专家们十分关注的问题。电气故障一旦出现，电网有可能会发生无法逆转的变化而进入不稳定的状态，或者会在经过波动后回复到稳定的状态。电网是不是能在故障触发后保持稳定是一个很重要的标准。

任务二，回答在何种情况下电网可能会进入不稳定状态。不稳定的电网可能会有不同的表现形式，即不同类型的不稳定性在时变多变量数据集上可能会表现出不同的特征。分析师期望找出是何种故障（母线接地、交流线路断开或发电机退出等）导致了当前的电网不稳定状态。

能够辨别电网在何种进入不稳定的状态。不稳定性有着不同的类型，而这些不同的类型在时变多变量数据集上会展现出不一样的特征，这些就使得不稳定的电网展现出不同的表现形式。合作者的期望是找出什么类型的故障导致了电网不稳定状态的出现。

任务三，找出电网不稳定状态下的典型模式。在确定故障类型后，分析师希望进一步地调查电网失稳状态下的典型特征（如周期性失稳或单摆失稳）。这些特征不仅可以用来对电网进行不稳定类型分类，同时可以帮助分析师更好地理解电网的失稳状态。

找出典型的电网不稳定状态。这是在确定故障的类型后，合作者们期望可以对此找出电网不稳定状态下的经典特征（如周期性失摆或者单摆失稳）。这些特征都可以用来对电网进

行不稳定类型分类，也可以来帮助分析时了解电网的不稳定状态。

任务四，找出导致电网失稳的具体原因，如故障出现在具体哪段母线上等。本文合作者对电网不稳定的情况推理表了现出强烈的兴趣。了解电网变得不稳定的原因不仅有助于采取相应的措施，从而减轻电力故障带来的成本，也有助于防止这种问题再次发生。

找出导致电网出现不稳定的具体原因。例如故障出现在具体哪个部分等。文章合作者更希望可以针对电网的不稳定进行推理。这些有助于针对不稳定的原因采取合适的措施，进而减轻电力故障造成的损失，也有助于防止这种问题再度出现。

为完成上述任务，本文作者与合作者讨论，总结出了六个要求：

要求一：描述整个电网的拓扑结构。应该给出电网及其配置的概述，以支持对总体结构和模式的识别。

能对整个电网进行描述的拓扑结构。给出电网及其配置的大体总览，方便对于总体结构和模式的认知。

要求二：刻画整个电网的运行状态。计算出的时变多变量数据预示着全网状态的演变，一个合理的交互界面有助于展示并鉴别全网变化趋势及每个变量变化的细节。

对于整个电网的运行状态进行刻画。计算出的时变多变量数据代表了电网状态的演变，需要一个合理的交互界面帮助鉴别和展示电网数据的变化趋势，以及变量变化的相应细节。

要求三：异常检查。异常的值（某一时刻的值）或模式（某个时期内值的变化）很可能表征了电网的不稳定性。因此，自动异常值检测算法对于支持异常值的有效识别是必需的。检测到的异常应该通过独特的视觉编码自动高亮，以便分析人员能够根据其领域的专业知识和经验发现异常模式。

异常的检查。异常值的变化代表了电网的不稳定性。因此必须要有一个有效识别异常值的自动异常检测算法。

检测到的异常通过独特的视觉编码提供高亮，以便于分析人员可以根据经验和相关领域的知识发现异常模式。

要求四：检测多个变量的时序数据。不同变量的值会随着时间的推移而变化，这点在发生大型故障时尤其明显。系统应该支持在某个时间点或某个时间段内对变量进行逐一观察，以便分析人员能够专注于更详细的特征变化。而且，视觉编码和设计应该能够说明不同变量的周期性变化和异常模式。

对于多变量的时序数据的检测。不同变量的值会随着时间的推移而发生变化。在发生大型故障时最为明显。系统需要支持在某个时间点或者时间段对于变量进行逐个观察，使得分析人员可以更专注与详细的特征变化。而且，视觉编码以及设计需要能够表现出不同变量的异常模式和周期性变化。

要求五：设计成对时变序列的可视化方法以提供观察“时序多变量数据对异常演变的影响”的直观手段。除了研究个体变量的变化之外，对变量对（如电压和频率）的比较和匹配检测对于不稳定性的分析具有重要价值。

设计出对时变序列进行可视化的方法，使得可以直观的观察“时序多变量数据对异常演变的影响。”除了研究单个变量的变化以外，对变量对（例如电压与频率）的比较以及匹配

检测也有很重要的价值。

要求六:显示电网存在的局部扰动。一旦电网波动后稳定,大部分特征值趋于保持不变。然而,在整个电网保持稳定的情况下,多个节点的频率扰动仍然存在。因此,除了揭示不同变量的整体变化外,还应展示电网内存在的局部扰动。

对于电网存在的局部扰动进行展示,一旦电网波动稳定后,大部分的特征值趋于不变。但是同时,多个节点的频率扰动仍然还在。因此,除了展示不同变量的整体上的变化,还要展现电网内的局部扰动。

系统设计 with 实现

本系统采用浏览器-服务器 (B/S, browser and server) 架构。B/S 架构与客户端-服务端 (C/S, client and server) 架构不同,它具有相对平台独立、易用和易开发的特点,这些特点分别体现于: B/S 架构的终端运行在浏览器上,软件运行的最终效果与操作系统较不相关;软件的使用成本低,用户只需处在能对服务器发起会话的网络中借助浏览器即可访问系统,无须引入可能更为复杂的环境配置等环节; JavaScript, Python 是目前比较流行的语言,它们背后强大的社区能为使用这门语言的开发者提供迅捷有效的支持,可以更为快捷的实现想要的效果。

依据数据通路和各模块间的依赖关系,可将本文系统划分为四个模块:暂态稳定计算模块、数据预处理模块、后端服务器和前端浏览器。本章会对这些模块的具体设计进行介绍。

4.1 暂态稳定计算模块

暂态稳定计算模块可以生成电网仿真数据,该模块产生的数据包括但不限于:记录每个节点(母线)静态属性的潮流母线数据文件、记录每条边(基准电压不同的母线间的连线)静态属性的交流线和变压器数据文件、定义故障表现方式(故障类型、出现的时刻等)的网络故障数据文件、定义结果文件格式的暂态稳定计算结果元数据文件以及暂态稳定计算结果文件。

4.2 数据预处理模块

数据预处理模块针对原始仿真数据中存在的数值重复、数据缺失等问题,首先对数据进行清洗,再从潮流母线、交流线和变压器数据文件抽取出电网拓扑图。Dijkstra 算法常用于解决不含负权环的单源最短路问题,本文作者实现了变种的 Dijkstra 算法 5 用于解决电网系统中多故障源的“多源”最短路问题。进一步的,本模块根据暂态稳定计算结果元数据文件,解析了暂态稳定计算结果文件,为不同的物理量生成了不同的 Json 文件,存储在服务器端以供用户按需访问。最后,根据网络故障数据文件,本模块可以计算故障发生后每段母线上任一物理量与故障前一时刻的差分,这些差值所形成的差分文件将在生成差分 WaveLines 图时发挥作用。

当两段母线间,存在多条线路直接或间接地将它们相连时,它们间的并联组框可以通过如下公式算得,或者近似的,可以用多条道路中阻抗最小的一条支路阻抗作为近似值。

4.3 后端服务器

后端服务器负责用户验证与授权、-应用用户的资源请求，并根据用户指定的统计质量控制方法及给出的参数，在线计算异常区间。后端服务器是由 Django 框架实现的。Django 是一款免费的、开源的高级 Python Web 框架。它由一批经验丰富的开发者构建，能够代替它的使用者处理许多在 Web 开发过程中遇到的麻烦，使得开发者可以专注于编写应用程序本身，快速迭代产品。

4.3.1 用户验证

用户验证的目的是在于验证用户是否确实是其所生成的角色，用户授权的目的在于定义验证用户的资源访问权限。为了简单起见，我们在后文中将用用户验证去指代这两种任务。一个用户验证系统应包含：用户、权限（一个用于标识特定用户能否访问某种特定资源的标识）、用户组、用于登录或限制访问内容的表单和其他视图工具以及一个可定制的密码哈希系统。Django 中的验证系统采用了非常通用的设计，有些 Web 验证系统中常见的有些特性并没有被其支持。相反的，这些常见问题的解决方案可以从第三方库中获得。这些常见问题有：密码强度检查（检查密码长度、是否含有不同类型的字符等）、最大试错次数和基于第三方平台的验证方式等。

4.3.2 RESTful 资源请求架构

当下是一个各种智能终端（如手机、平板、桌面电脑和其他专用设备）如雨后春笋般涌现的时代，一套规范的应用程序接口（Application Program Interface, AP）命名方式可以很好地帮助前后端通信的设计与实现，使得应用开发者可以快速、正确地开发程序，并提高程序的可维护性。Fielding 提出表现层状态转换（Representational State Transfer, REST）原则，定义了超文本传输协议（Hyper Text Transfer Protocol）请求中的资源（Resources）、表现层（Representation）和状态转换（State Transfer）。本文系统的 API 设计具有结构清晰、符合标准、易于理解、扩展方便的特点，是一个可以很好地相应用户（前端）请求的 RESTful 架构系统。

4.3.3 缓存算法

当后端服务器接收到用户请求时，不失一般性的，后端服务器将通过磁盘 IO 访问特定的样本数据。在计算中，高速缓存（Cache）是存储数据的硬件或软件组件，因此可以更快地为将来的数据请求提供服务；存储在缓存中的数据可能是早期计算的结果，或者存储在别处的数据的副本。当缓存满了时，算法必须选择丢弃哪些条目（Entry）来为新的缓存留出空间。高速缓存算法（Cache Algorithm，也经常称为高速缓存替换算法或高速缓存替换策略）可以用来优化计算机指令或算法，使得计算机中的硬件结构或程序能更好地管理、使用高速缓存。

为保证可以高效、低延迟地响应用户请求，本文作者在算法 6 中实现了一个基于最近最少使用策略（Least Recently Used, LRU）的缓存管理功能来减少系统对磁盘的访问次数。该实现的特点在于调用 get 和 put 函数的时间复杂度都是 $O(1)$ 的。这是由于，在给定迭代器指定插入删除的位置时，链表的插入和删除的时间复杂度都为 $O(1)$ ；访问通过哈希算法实现的映射表的时间复杂度同样是 $O(1)$ 的。

4.3.4 大规模电网母线聚合算法的优化

假设母线数量为 M ，时刻数量为 T ，算法 4 中 Agglter 函数的时间复杂度为 $O(MT)$ 。考虑到用户在探索大规模电网的过程中会频繁地拖动左侧的滚动条，每一次拖动前端都将向后端服务器发送 AJAX 请求[48]。倘若后端服务器每收到这么一次请求都需要耗费 $O(MT)$ 的时间复杂度来调用 Agglter 无疑是十分耗时的。为此，本文作者设计了一种空间换时间的策略，

以 $O(MT)$ 的空间复杂度和 $O(MT)$ 时间复杂度的预计算, 将每次后端服务器处理 AJAX 请求的时间复杂度由 $O(MT)$ 变为了 $O(T)$ 。

4.4 前端模块

4.4.1 模块构建工具

前端模块负责快速绘制可视化结果, 提供交互界面等任务。本模块引入了 Webpack[49] 和 Babel[50], 在程序开发和程序发布两个环节都为软件开发者提供了极大的帮助。Webpack 工具不仅支持通过 dev-server 监听 HTML 源文件的改动对网页进行热更新, 还可以将多个 JavaScript 文件打包成一个 JavaScript 文件。

Babel 作为 JavaScript 编译器, 紧跟时代潮流, 支持软件开发者使用 ECMAScript 2017(ES8) 语法。上述特性不仅可以帮助软件开发者更快地开发和发布产品, 同时发布产品时仅需向外界提供被 Babel 解释过的代码, 很好地保护了源代码。

在开发现代软件的过程中, 开源社区不仅使用 ES8 新标准, 还采用了一系列拓展的 JavaScript 语法来加强程序语言的表达能力。为适应这一潮流, 本模块还采用了如 babelplugin-transform-object-rest-spread 等的 Babel 相关插件, 用以支持如对象解构赋值等的高级语言特性。

4.4.2 交互组件库

为实现具有丰富交互特性的前端模块, 本文系统采用了 React, D3 等前端交互组件库, 支持用户在 SVG 或 canvas 元素上完成探索任务。

4.4.2.1 React

React 由 Facebook, Instagram 及由个人开发者所组成的社区所维护, 是一种用于构建用户交互界面的 JavaScript 第三方库。它具有如下的特点:

- 声明式: 使用 React 创建交互式用户界面十分轻松, 它能为应用程序中的每个状态设计简单的视图, 并将在数据更改时有效地更新和渲染正确的组件。声明式编程可以使得代码更具可预测性, 更易于调试;
- 组件化: 用户可以构建具有自我状态管理能力的封装组件, 然后将其组合成复杂的用户交互界面。由于组件逻辑是用 JavaScript 而非模板编写, 因此可以通过应用程序轻松传递丰富的数据, 并将状态保留在文档对象模型 (Document Object Model, DOM) 之外;
- 一次编写多次使用: 类比的, React 仅作为传统的“模型-视图-控制器” (Model, View, Controller) 架构中的视图层, 并不对所处系统的技术栈做出假设。因此, 用户可以在不重写现有代码的前提下, 通过 React 开发新功能。同时, React 也可以和 Node.js 结合支持服务器端渲染, 或通过 React Native 赋能移动应用程序。

4.4.2.2 D3

D3 作为

JavaScript 第三方库, 主要用于进行在 Web 浏览器中的交互式数据可视化编程, 历经多个版本的改动, 目前最为主流的为 D3 v4 版本 (第四版本)。它遵循主流的 SVG, HTML5 和 CSS 标准, 并支持对视觉效果的高度定制化编程。截至今日, 已有数以万计的网站使用它作为构建可视分析系统最常用的工具, 其作者 Mick Bostock 的画廊[54] 展示了诸多使用 D3 构建的样例。同时, 许多鼎鼎有名的第三方开源库也直接使用或借鉴了 D3 的代码或设计理念, 这些软件包括但不限于: 百度的 ECharts、蚂蚁金服的 G2、Vega 及其拓展 Vega-lite。本文系统遵循业界最佳实践, 采用 D3 实现了电网拓扑图、统计图、WaveLines 视图及半矩阵视图。

4.4.3 状态管理器

Redux 是为 **JavaScript** 应用开发的状态容器。它非常轻量，连同其依赖库，仅有 **2KB** 大小。它可以帮助开发者编写在不同环境（客户端、服务器和本地）中运行一致并且易于测试的应用程序。例如，它通过支持“结合时间旅行（**time travelling**）调试器”的实时代码编辑器提供了很好的开发者体验。

根据 **Redux** 的设计理念，本文作者定义了全局状态用来描述当前应用所包含的服务器响应、缓存数据和尚未发往服务器的局部数据，将其与 **React** 一起配合使用，构建了复杂的单页面应用。